AP

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/840,727	04/23/2001	Alando M. Ballantyne	014208.1360	4537

35005 7590 01/19/2006

BAKER BOTTS L.L.P.
2001 ROSS AVENUE, 6TH FLOOR
DALLAS, TX 75201

EXAMINER	
STEELMAN, MARY J	
ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 01/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/840,727	**Applicant(s)** BALLANTYNE ET AL.	
	Examiner Mary J. Steelman	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 October 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23 and 25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) *1-6 and 13-22* is/are rejected.

7)☒ Claim(s) *7-12,23 and 25* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 April 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to Amendments and Remarks received 14 October 2005. The prior Non Final Office Action is hereby withdrawn. Per Applicant's request, claim 20 has been amended. Per Applicant's request, 4/22/2005, claim 24 has been canceled. Claims 7-12, 23, and 25 have been previously noted as containing allowable subject matter (Office Action 12/22/2004). Claims 1-23 and 25 are pending.

Response to Arguments

2. Applicant's arguments with respect to claim1-6 &13-22 have been considered but are moot in view of the new grounds of rejection.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-6, 13-18, and 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,810,429 B1 to Walsh et al., in view of "What is the Document Object Model?", by Jonathan Robie (hereinafter Robie) (1998).

Per claim 1:

A method for outputting data from a legacy computer system, the data output in Extensible Markup Language format, the method comprising:

(Walsh: Col. 15, line 61 – col. 16, line 10, "During the design process, one specifies what information is to be retrieved from the database and how to map the information into an XML document.")

-generating a model of the legacy computer system, the model comprising one or more incidents within one or more applications that output data;
(Walsh: Col. 11, lines 10-30, update, delete, add (incidents), col. 15, line 61 – col. 16, line 10, generating a model.)

-mapping the model of the legacy computer system to an Extensible Markup Language schema;
(Walsh: Col. 15, line 61 – col. 16, line 10, "...map information into an XML document...", when generating a model.

-based at least on the mapping of the model of the legacy computer system to the Extensible Markup Language schema, automatically modifying the one or more applications of the legacy computer system that output data, the one or more modified applications operable to output data written using a Document Object Model from the legacy computer system in Extensible Markup Language.
(Walsh: col. 4, lines 2-5, "These basic interfaces allow our integration system to view, modify and interact with linked legacy applications or legacy data sources.", col. 4, lines 36-39, "The design tools 140 support the definition of XML document formats. The design tools also allow us to define mappings of the SML document formats and the legacy data formats...", col. 6, lines 10-13, "DOM

(Document Object Model), is a programming interface specification that specifies a tree which applications may then explore or modify.", col. 6, lines 40-46, The 'add' operation inserts a new document supplied in the form of a DOM.)

Walsh disclosed (col. 3, lines 53-54) "XML as a universal data encoding and interchange format." Col. 3, line 67- col. 4, line 4, "XML supports links that point to multiple documents...These basic interfaces allow our integration system to view, modify and interact with linked legacy applications of legacy data sources." Walsh failed to specifically disclose output data written using a Document Object Model from the legacy computer system in Extensible Markup Language. However, note the Robie reference (first paragraph) defining Document Object Model. It is an application-programming interface (API) for HTML and XML documents, defining the logical structure of documents and the way a document is accessed and manipulated. XML presents data as documents and the DOM is used to manage the data. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to consider the XML support and interfaces as disclosed by Walsh to include DOM. Walsh recognized the need (col. 2, lines 56-60) "to convey transactional data between any number of databases regardless of their format, context, and access methodology."

Per claim 2:

-providing the legacy computer system with a writer engine, the writer engine having the Extensible Markup Language Schema loaded as a data file;

(Walsh: Col. 10, lines 1-13, legacy systems mapped to XML, converting results (writer engine) and delivering in a custom format to a user.)

-calling the writer engine with the modified one or more applications when the one or more applications output data, the writer engine populating the Document Object Model according to the Extensible Markup Language schema by building a Document Object Model instance with one or more contexts.

(Walsh: Col. 10: lines 4-20. As an example, Walsh disclosed "A query operation to a relational database involves extracting the schema of the database by generating a SQL runtime access component (RAC) (an XML interface) which makes calls to the database, converting the resulting data into the XML format, and handing the XML to an agent (an XML interface). The agent delivers the XML to the front-end for conversion using the style sheet (a writer engine operation). See claim 1 above regarding a DOM as an XML interface.)

Per Claim 3:

-applying one or more XSLT stylesheets to restructure the Document Object Model instance for outputting data in a predetermined format.

(Walsh disclosed (col. 4, lines 22-27) a "front-end interface 120...to present information to users 103 using standard presentation methodologies The front-end interface also allows the user to modify information and to generate transactions to initiate enterprise processes or workflow..." Col. 10, lines 10-13, "The agent delivers the XML to ;the front-end 120 for conversion to the HTML form 121 using the style sheet 126 so that the data can be viewed by the user 103 using a standard browser 124." See FIG. 1b, #126.)

Per claim 4:

A system for outputting data from a legacy computer system in an Extensible Markup Language format, the system comprising:

-a modeling engine in communication with the legacy computer system, the modeling engine operable to generate a model of outputted data written by an application residing on the legacy computer system;

(Walsh: Col. 15, line 62- col. 16, line 10 – modeling engine, retrieves information from the data sources (outputted data written by an application residing on the legacy computer system), col. 4, line 46- Legacy system equivalent to 'data source'.)

-a mapping engine in communication with the modeling engine, the mapping engine operable to generate a modification specification by mapping the model of the outputted data written by the application residing on the legacy computer system to an Extensible Markup Language schema;

(Walsh: Col. 16, line 2, "map the information into an XML document."

-a code generation engine in communication with the mapping engine and the legacy computer system, the code generation engine operable to modify code of the application residing on the legacy computer system, based at least on the generated modification specification, to directly output data from a Document Object Model as Extensible Markup Language.

(Walsh: Col. 6, lines 10-13-DOM as a programming interface specification, col. 6, lines 30-46, modify code using update, delete, add operations. Col. 6, lines 7-8, Output data from a DOM is in XML form.)

Per claim 5:

-a context table associated with the legacy computer system, the context table providing the Extensible Markup Language schema to the legacy computer system;

(Walsh: Col. 9, lines 65-67, "design tools use the service bridge to extract schemas from the data sources", col. 11, lines 32-37, "mapping design tool extracts the schema from legacy databases. Because schema extraction is an expensive and time consuming task, the tools allow one to save extracted schemas (context table) on a disk for subsequent use. Also, see FIG. 3, data source / legacy is stored in #111, providing mapping to XML. See Table 1, col. 11.)

-a writer engine loaded on the legacy computer system and having the Extensible Markup Language schema stored as a data file, the writer engine communicating with the modified legacy computer system application to buffer data in plural contexts within the Document Object Model for output as Extensible Markup Language.

(Walsh: FIG. 7, #750, #755, Col. 16, lines 11-39, Update, delete, add/create modify legacy applications. See FIG. 5. Col. 4. line 17, Back-end interfaces to the legacy system.)

Per claim 6:

-the writer engine is coded in the computer language of the legacy computer system.

(Walsh: Col. 4, lines 17-2, Note the writer engine for the 'back-end' legacy system.)

Per claim 13:

This is a system version of claims 1 & 2 noted above.

See rejection of limitations as addressed in claim 1 & 2 above.

Per claim 14:

-the writer engine populates a Document Object Model as a schema element aligned with a current one of the plural contexts by creating Extensible Markup Language tagged nodes down through the schema element of the output data if the schema element of the output data is a descendant of the current context.

(Walsh: FIGs. 7 & 9, col. 16, lines 40-63, The execution flow for basic database access requests is shown in FIG. 7. A request is received, the caller & document identity is determined, the mappings construct the SQL statements to retrieve and execute the statement. The result set is 'walked' (plural contexts, tagged nodes down through the schema), fields are extracted to build the XML document.)

Per claim 15:

-the writer engine is further operable to determine a minimal mutual ancestor of the schema element of the output data and the current context and to traverse the Extensible Markup Language tagged nodes for the current context up to the minimal mutual ancestor and to create Extensible Markup Language tags for the schema element of the output data down from the mutual ancestor.

(Walsh: Col. 3, lines 62-65, "XML enables us to create customized 'tags' that provide functionality …", col. 10, lines 30-33, Document Type definitions (DTDs) are used to generate the output to user interfaces. Also see examples of tags in code segments shown in columns 7 & 8.

See rejection of claim 14 above, regarding traversal of XML tagged nodes and ancestors to provide output.)

Per claim 16:

See rejection of limitations addressed in claim 6 above.

Per claim 17:

-the application comprises a legacy computer system application modified to output an Extensible Markup Language schema element with the output data.

See rejection of limitations addressed in claim 1 above.

Per claim 18:

-the writer engine is written in the code of the legacy computer system.

See rejection of limitations addressed in claim 6 above.

Per claim 20:

A method for outputting data from a legacy computer system from a Document Object Model instance as Extensible Markup Language, the method comprising:

modifying an application of the legacy computer system such that the modified application is operable to output data having a schema element of a target Extensible Markup Language schema, the output data corresponding to a write operation of the application;

outputting data from the modified application, the output data having the schema element of the target Extensible Markup Language schema;

aligning the schema element of the output data and a current context;

writing the schema element of the output data to a current one of plural contexts of the target Extensible Markup Language schema; and

populating a Document Object Model with the output data to output an Extensible Markup Language instance.

See rejection of limitations addressed in claims 1 & 2 above.

Per claim 21:

-aligning the schema element of the output data further comprises:

-determining that the schema element of the output data is a descendant of the current context;

-creating appropriate Extensible Markup Language tags down through the schema element of the output data, each Extensible Markup Language tag down through the schema element of the output data being associated with an ancestor of the schema element of the output data.

See rejection of limitations addressed in claim 15 above.

Per claim 22:

-aligning the schema element further comprises:

-determining a minimal mutual ancestor of the schema element of the output data and the current context;

-traversing the Extensible Markup Language tags for the current context up to the mutual ancestor of the schema element of the output data and of the current context;

-creating the Extensible Markup Language tags for the schema element of the output data down from the mutual ancestor to the schema element of the output data.

See rejection of limitations addressed in claim 15 above.

5. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,810,429 B1 to Walsh et al., in view of "What is the Document Object Model?", by Jonathan Robie (hereinafter Robie) (1998), and further in view of US Patent 6,209,124 B1 to Vermeire et al.

Per claim 19:

The system of Claim 18 wherein the code comprises COBOL.

Walsh failed to disclose COBOL. However, Vermeire disclosed Cobol at col. 18, lines18-21, in his invention related to using mark-up language.

Therefore, it would have been obvious, to one of ordinary skill in the art at the time of the invention to modify Walsh / Robie, by including the teachings of Vermeire which disclose the use of Cobol in legacy systems, because Vermeire recognized the need (col. 4, lines 58-67) "to satisfy the demand for external system access to conventional system data and business logic, including markup language versions of the data and business logic…" without rewriting the application and without making huge demands on processor and computing resources. Likewise Walsh /Robie recognized the need to make legacy data available to users.

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman Mary Steelman
01/04/2006